Summary

- Model Parameters → Estimated from the data
- Model Hyperparameters → Can't be estimated from the data

Model hyperparameters are often referred to as parameters because they are the parts of the machine learning that must be set manually and tuned.

5 Epochs, Batches, Batch Sizes & Iterations

- Need these terms only if the dataset is large
- Break down the dataset into smaller "chunks" and feed those chunks to the neural network one-by-one

Epochs

When the ENTIRE dataset is passed forward and backward through the neural network only ONCE

We use multiple epochs to help our model generalize better

Batch & Batch Size

We divide large datasets into smaller batches and feed those batches into the neural network

Batch Size: Total number of training examples in a Batch

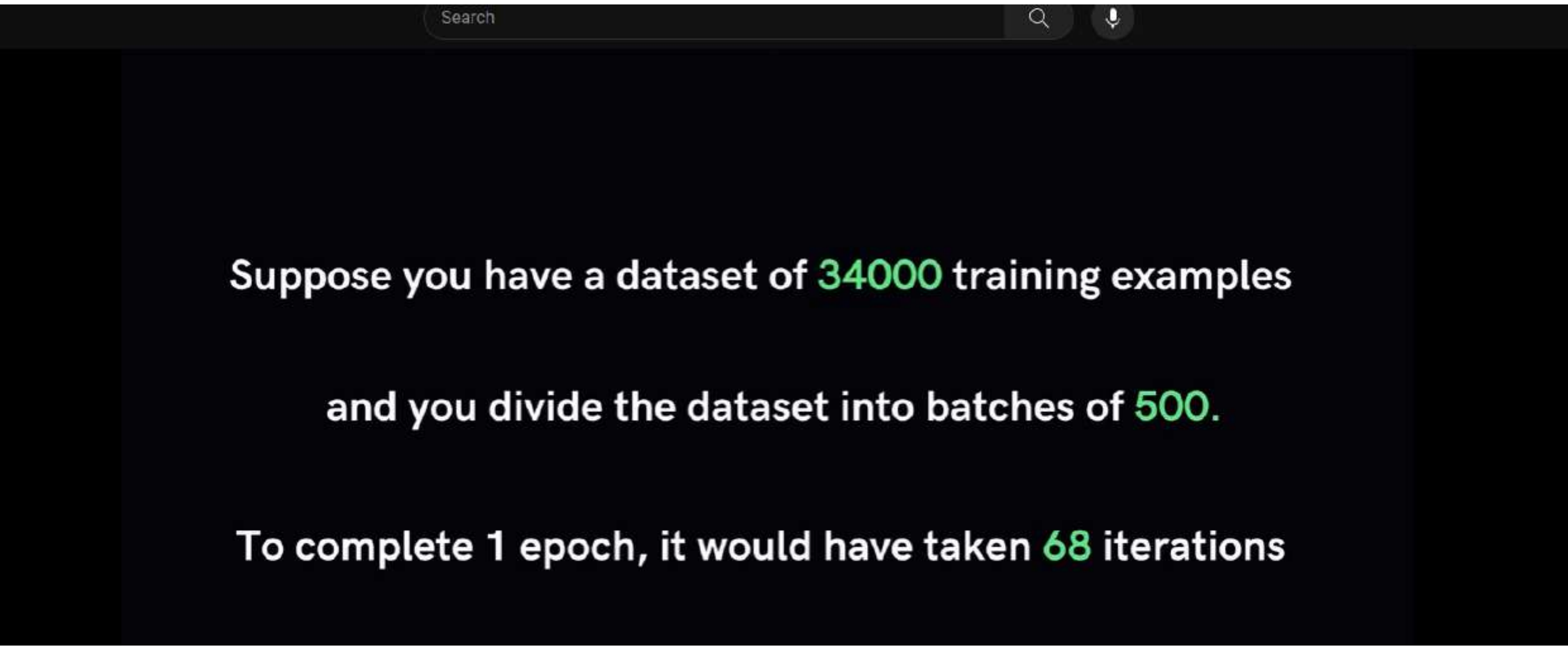
Neural Network Architectures

Iterations

Number of batches needed to complete one epoch

Alternatively,

Number of batches = Number of iterations for one epoch


Search
Suppose you have a dataset of 34000 training examples
and you divide the dataset into batches of 500.
To complete 1 epoch, it would have taken 68 iterations

Conclusion to Terms Used in NNs

How many hidden layers should I choose?

Which Activation function to pick?

Experiment, Experiment, Experiment!

Types of Learning

3 Main Types:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

① Supervised Learning

- Algorithms designed to learn by example
- Models are trained on well-labelled data

Each example is a pair consisting of:

- Input Object (typically a vector)
- Desired Output (Supervisory Signal)

During training,

SL algorithm searches for patterns that correlate with the desired output

After training,

takes in unseen inputs and determine which label to classify it to

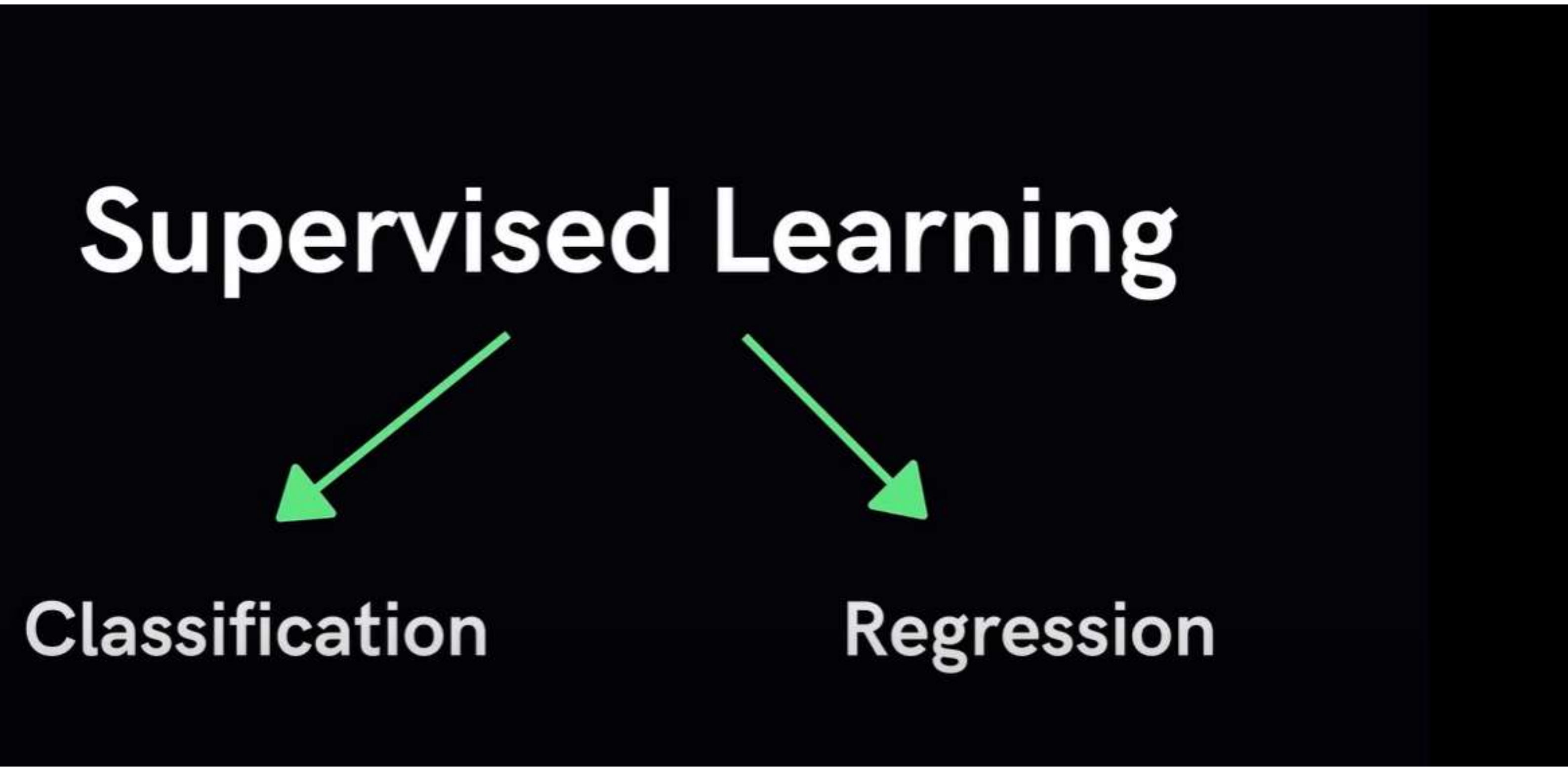
Objective of a Supervised Learning Model:

Predict the correct label for unseen data

$$y = f(x)$$

x = Input

y = Predicted Output


Supervised Learning
Classification
Regression

1 Classification

Take input data and assign it to a class/category

Example: Is email spam or not spam

Models finds features in the data that correlate to a class and creates a mapping function

This mapping function will be used to classify unseen data

Popular Classification Algorithms

1. Linear Classifers
2. Support Vector Machines
3. K-Nearest Neighbour
4. Random Forest

2 Regression

Model attempts to find a relationship between dependent & independent variables

Goal: To predict continuous values such as Test Scores

Equation for Basic Linear Regression

$$\hat{y} = w[0]*x[0] + w[1]*x[1] + \ldots\ldots + w[i]*x[i] + b$$

$x[i]$ = i^{th} Input Feature

$w[i]$, b = Parameters

Many Input Features

Popular Regression Algorithms

1. Linear Regression
2. Lasso Regression
3. Multivariate Regression

Applications of Supervised Learning

1. Bioinformatics
2. Object Recognition
3. Spam Detection
4. Speech Recognition

2 Unsupervised Learning

- Uses to manifest underlying patterns in data
- Used in exploratory data analysis
- Does not use labelled data, rather relies on the data features
- Goal: Analyze data and find important underlying patterns

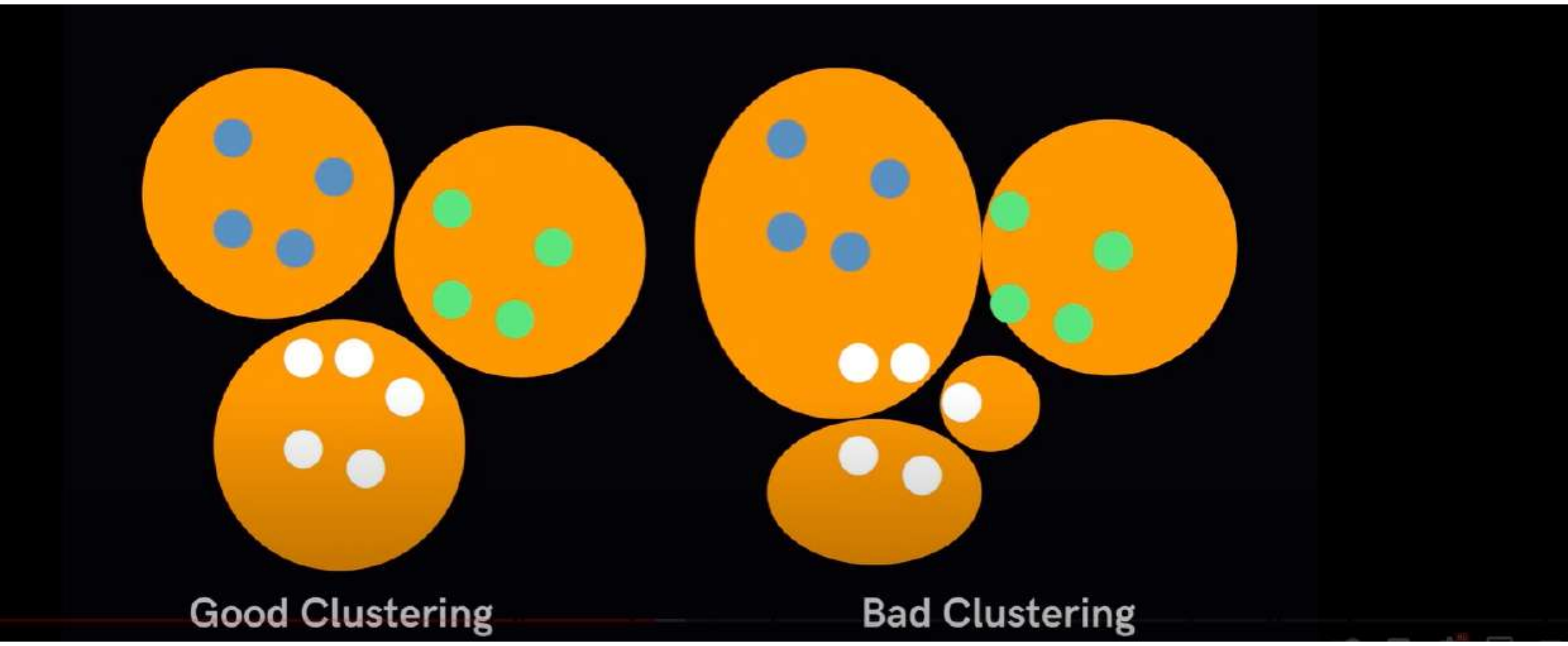
② Unsupervised Learning

- Uses to manifest underlying patterns in data
- Used in exploratory data analysis
- Does not use labelled data, rather relies on the data features
- Goal: Analyze data and find important underlying patterns

1 Clustering

Process of grouping data into different clusters or groups

Goal: To predict continuous values such as Test Scores


Good Clustering
Bad Clustering

Clustering:

1. **Partitional Clustering**

   Each data point can belong to a single cluster

2. **Hierarchical Clustering**

   Clusters within clusters

   Data point may belong to many clusters

Popular Clustering Algorithms

1. K-Means
2. Expectation Maximization
3. Hierarchical Cluster Analysis (HCA)

Search

❷ Association

Attempts to find relationships between different entities

Example: Market Basket Analysis

Applications of Unsupervised Learning

1. AirBnb
2. Amazon
3. Credit Card Fraud Detection

③ Reinforcement Learning

- Enables an agent to learn in an interactive environment by trial & error based on feedback from its own actions & experiences
- Uses rewards & punishments as signals for positive & negative behaviour

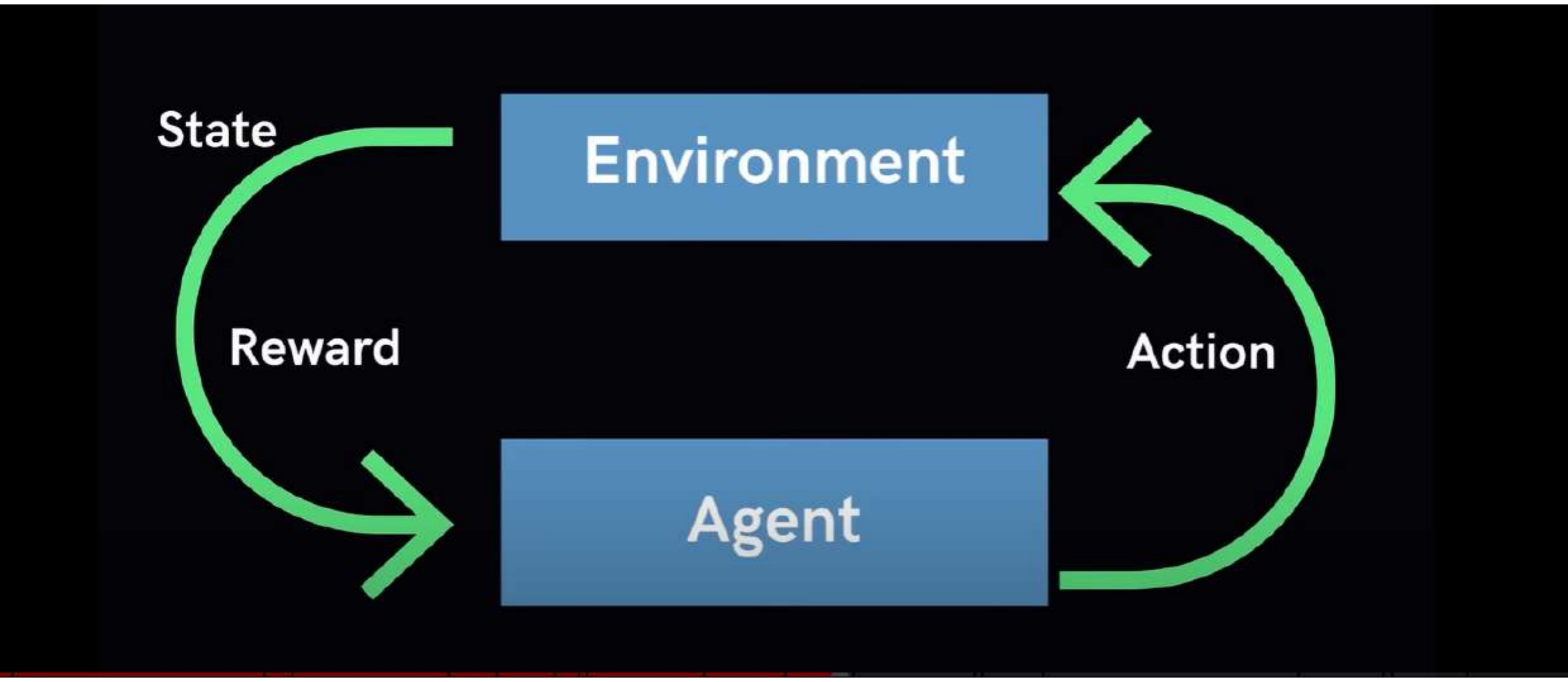
③ Reinforcement Learning

- Enables an agent to learn in an interactive environment by trial & error based on feedback from its own actions & experiences
- Uses rewards & punishments as signals for positive & negative behaviour
- Goal: Find a suitable model that would maximize the total cumulative reward

- Maximize the points won in a game over many moves

Penalized when they make wrong decisions
Rewarded when they make the right ones

Usually modelled as a Markov Decision Process


State
Environment
Reward
Action
Agent

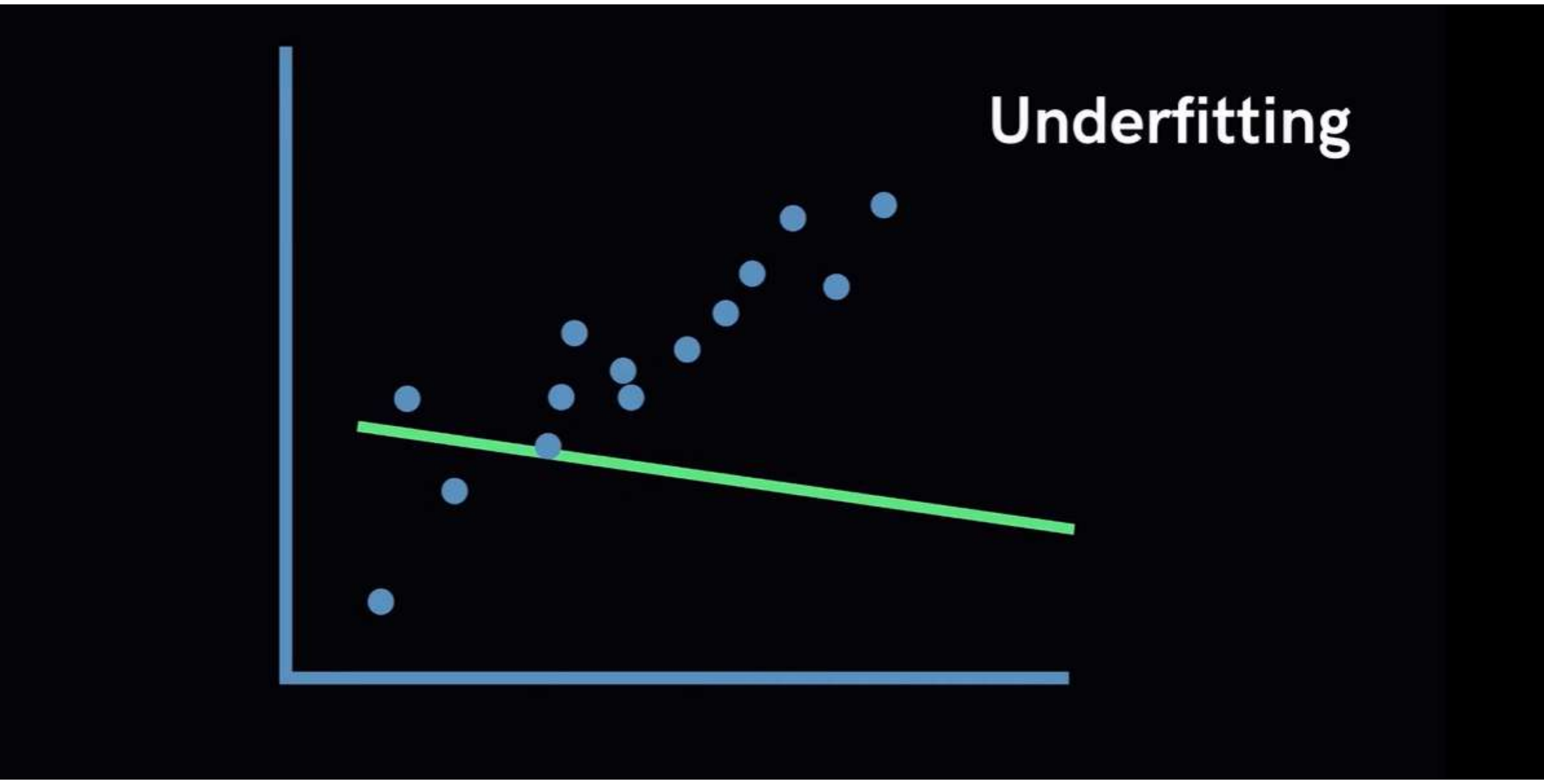
Applications of Reinforcement Learning

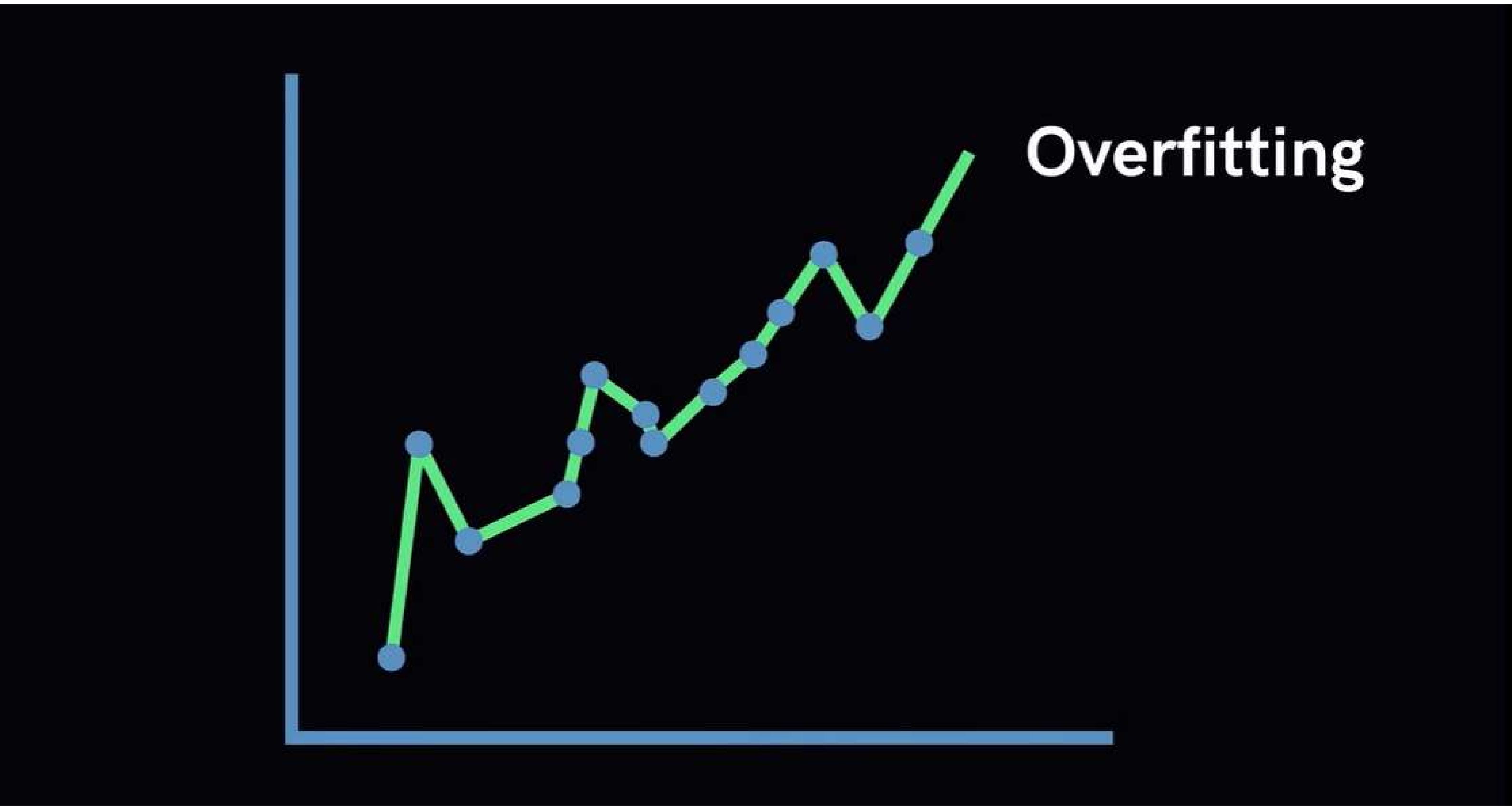
1. Robotics
2. Business Strategy Planning
3. Traffic Light Control
4. Web System Configuration

Core Problem in Deep Learning

Model should perform well on training data AND new test data

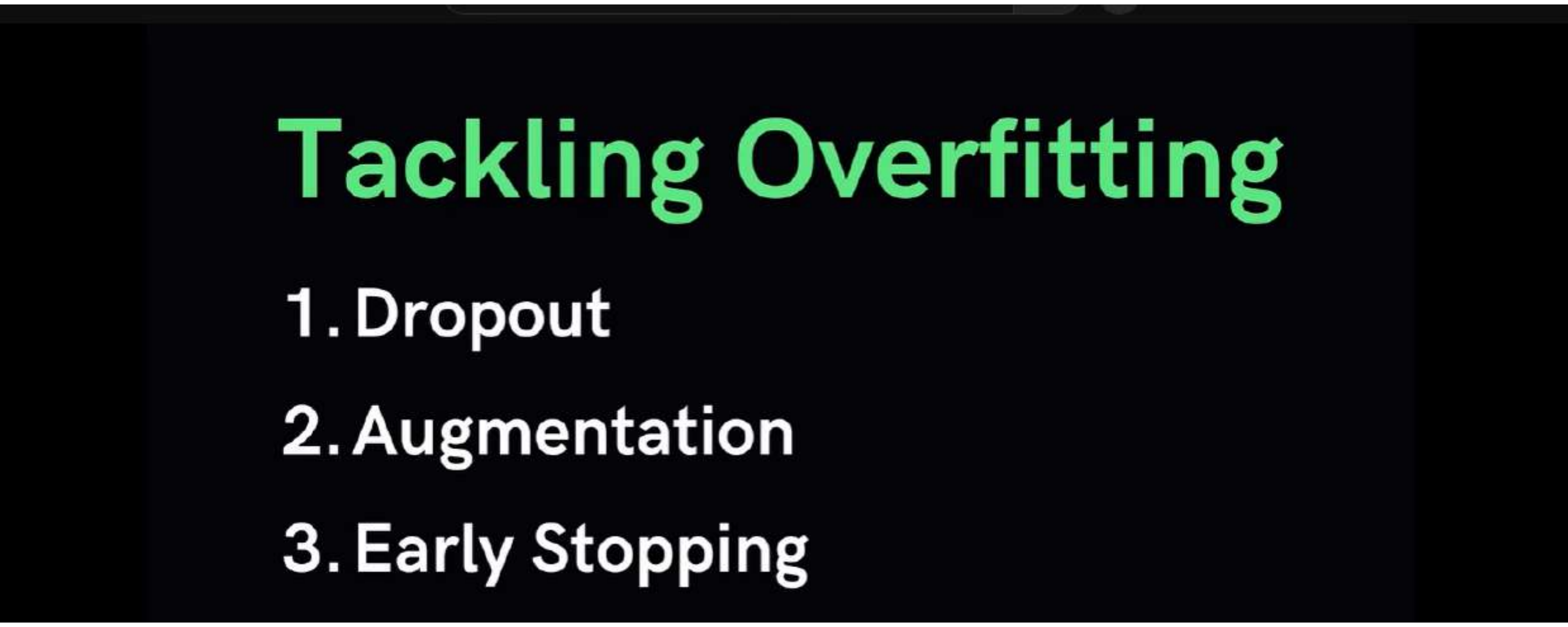
Most common problem faced is Overfitting

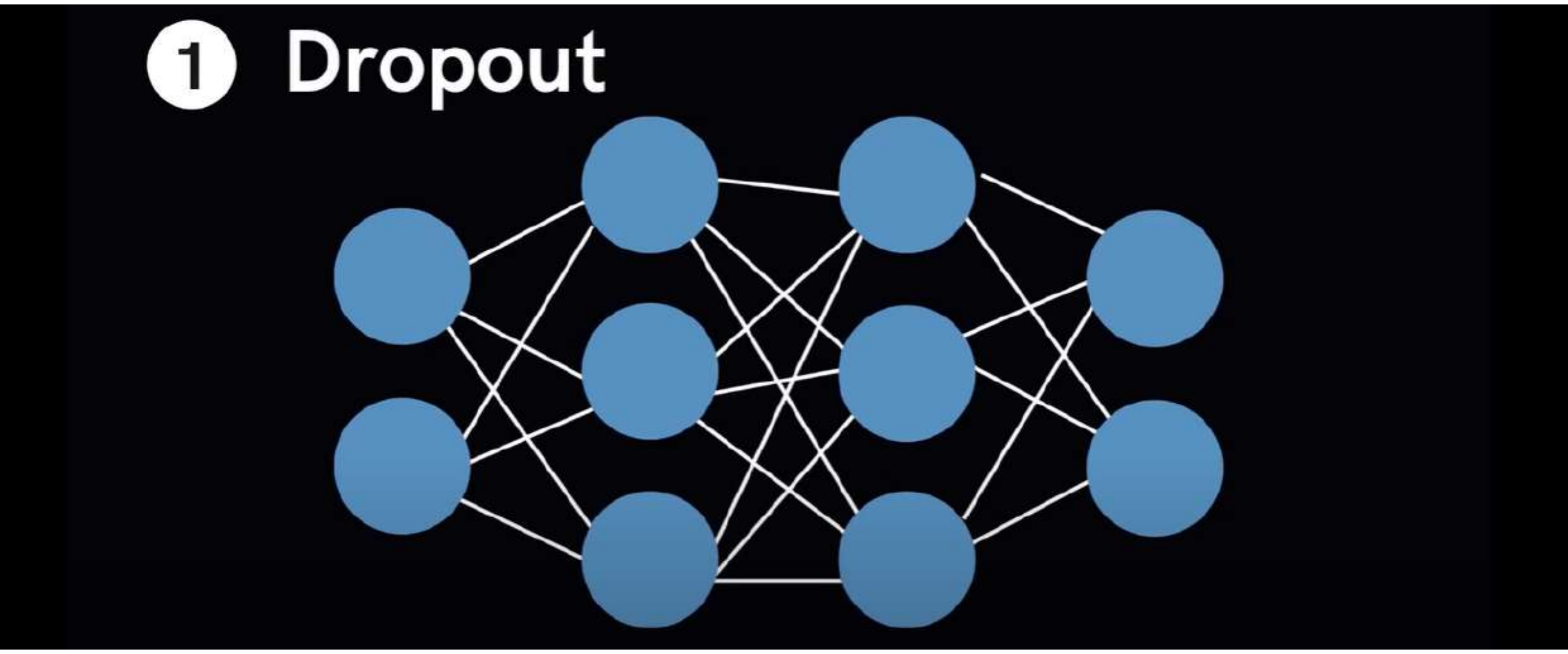

Underfitting

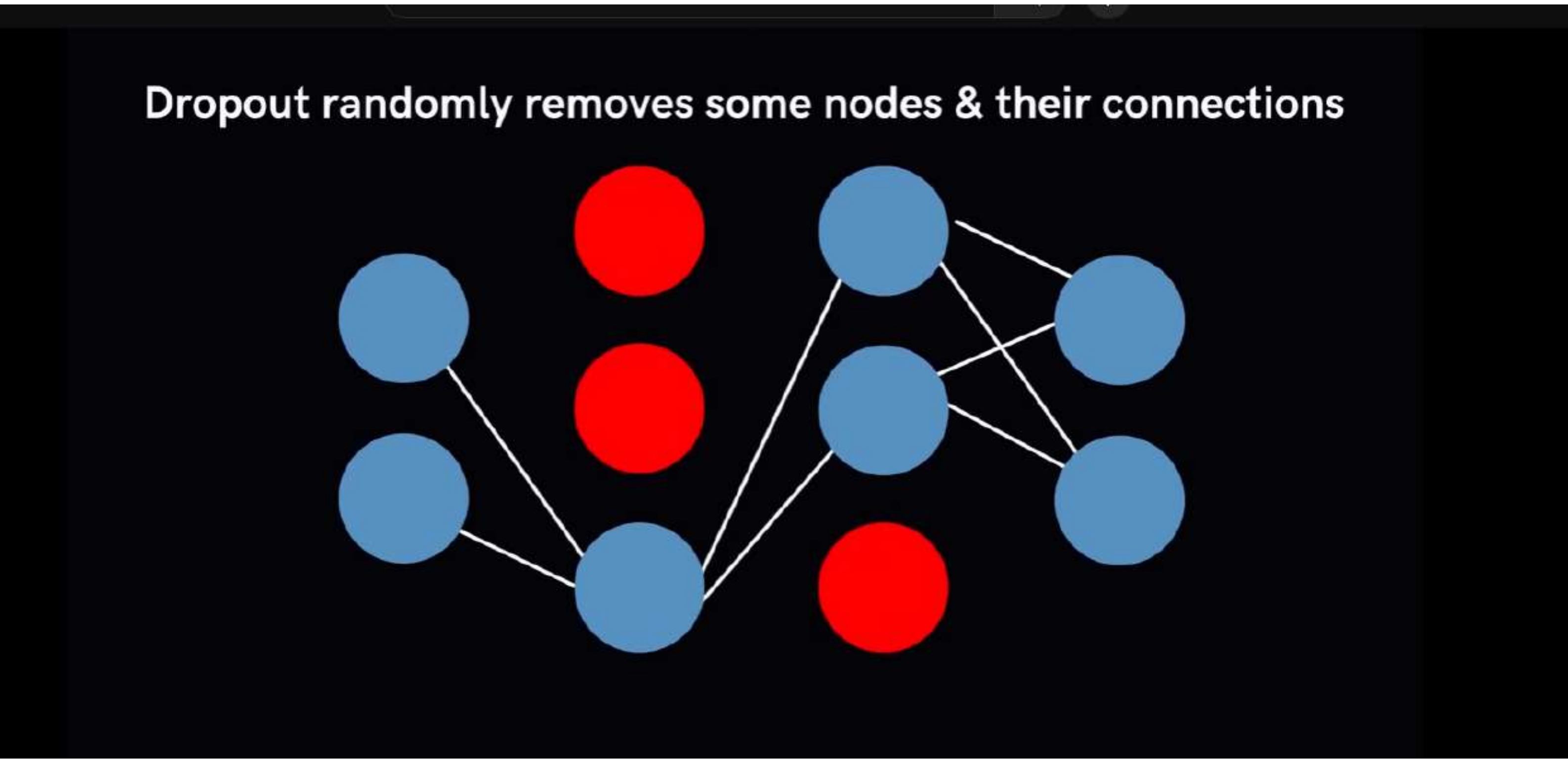

Overfitting

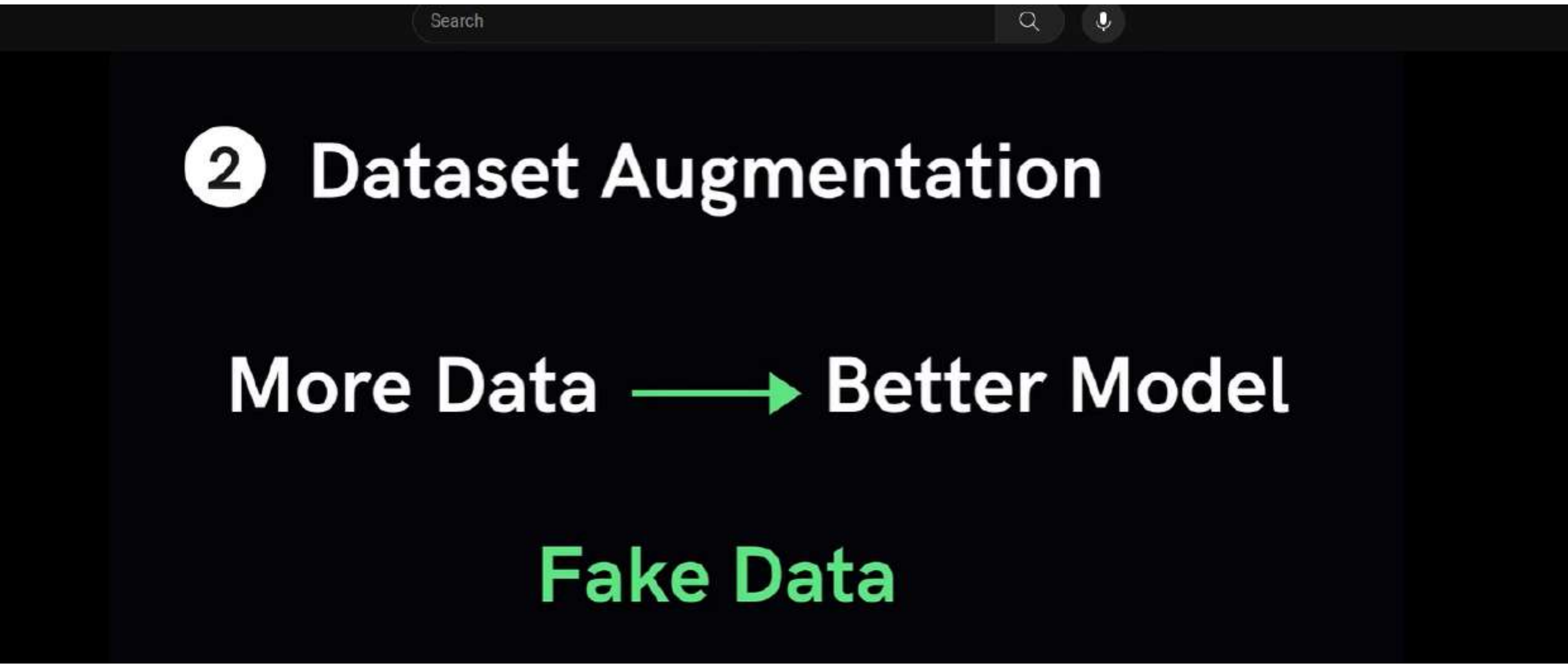

Not the best, but will perform
better on training data and
testing data

Tackling Overfitting


Tackling Overfitting
1. Dropout
2. Augmentation
3. Early Stopping


1 Dropout


Dropout randomly removes some nodes & their connections


Search
2 Dataset Augmentation
More Data
Better Model
Fake Data

2 Dataset Augmentation

Apply transformations on the existing dataset to get synthesize more data

③ Early Stopping

Training error decreases steadily

But, validation error increases after a certain point

Neural Network Architectures

Fully-Connected Feed Forward Neural Networks

Each neuron is connected to every subsequent layer with no backward connections

Neurons have activation functions

- Linear
- Sigmoid
- Tanh
- ReLU

Non-Linear

Inputs

Outputs

Hidden Layers

Neurons per hidden layer

Activation functions

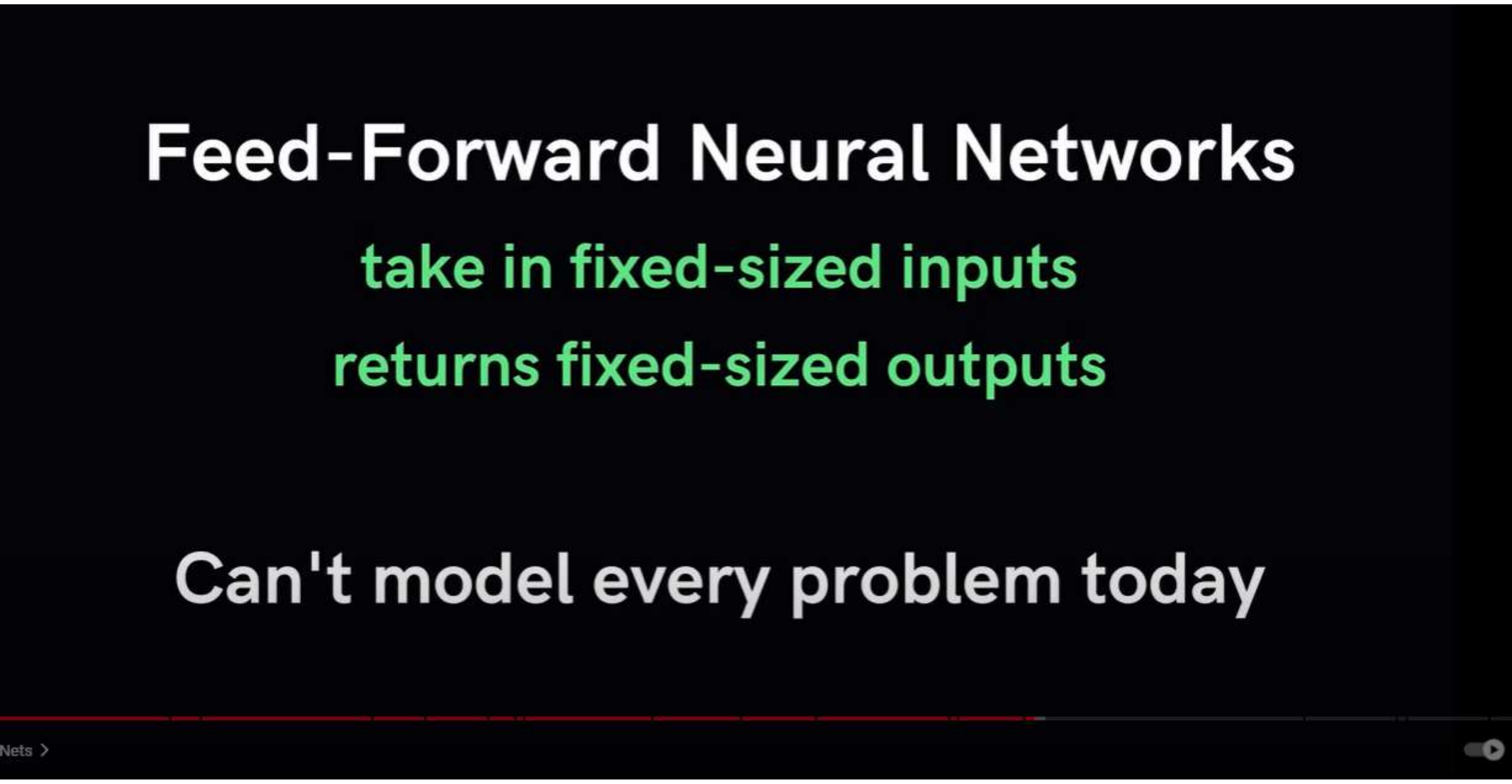

More Neurons
Larger Computational Resources

Recurrent Neural Networks

Feed-Forward Neural Networks

take in fixed-sized inputs

returns fixed-sized outputs



or Beginners

How does this apply to Neural Networks?

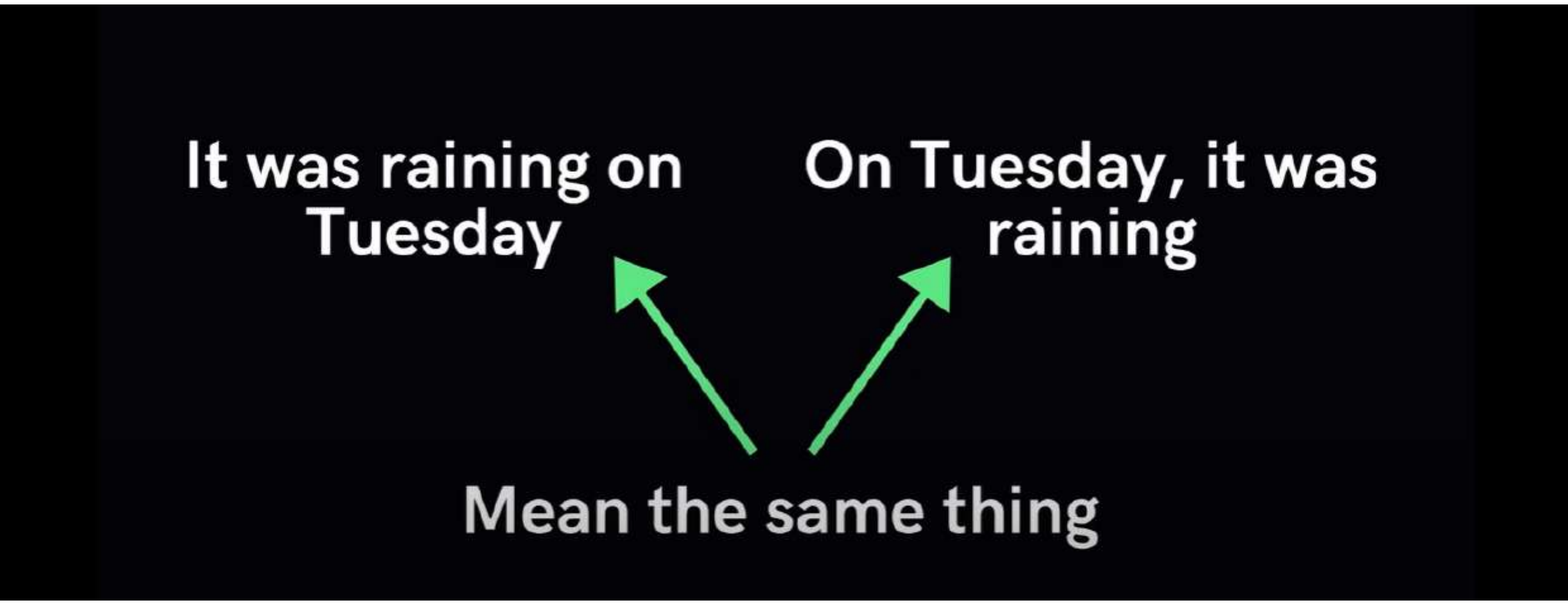
Information about the past must be supplied

Vanilla Neural Networks can't handle sequential data

Sequential Data is data in a sequence

Sequential Data is data in a sequence

Vanilla NNs don't share parameters across time


It was raining on Tuesday
On Tuesday, it was raining
Mean the same thing

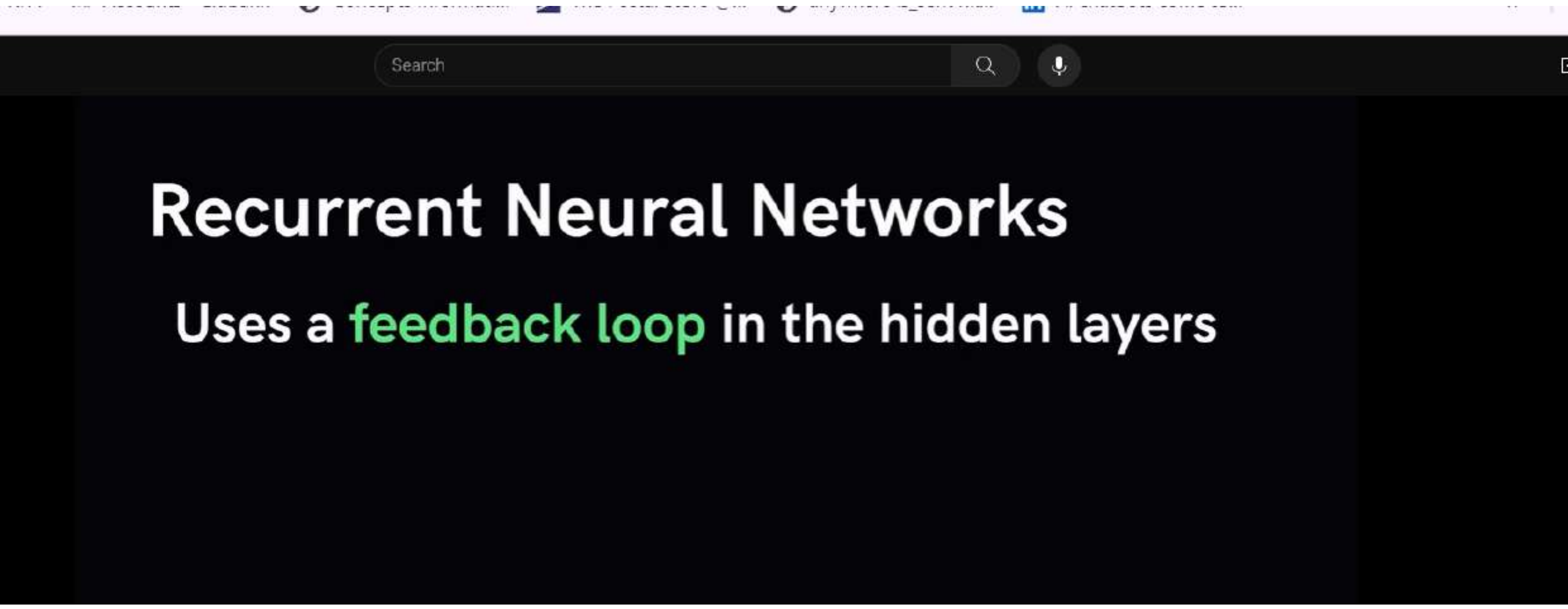
Sharing parameters gives the network the ability to look for a given feature everywhere in the sequence, rather than in just a certain area

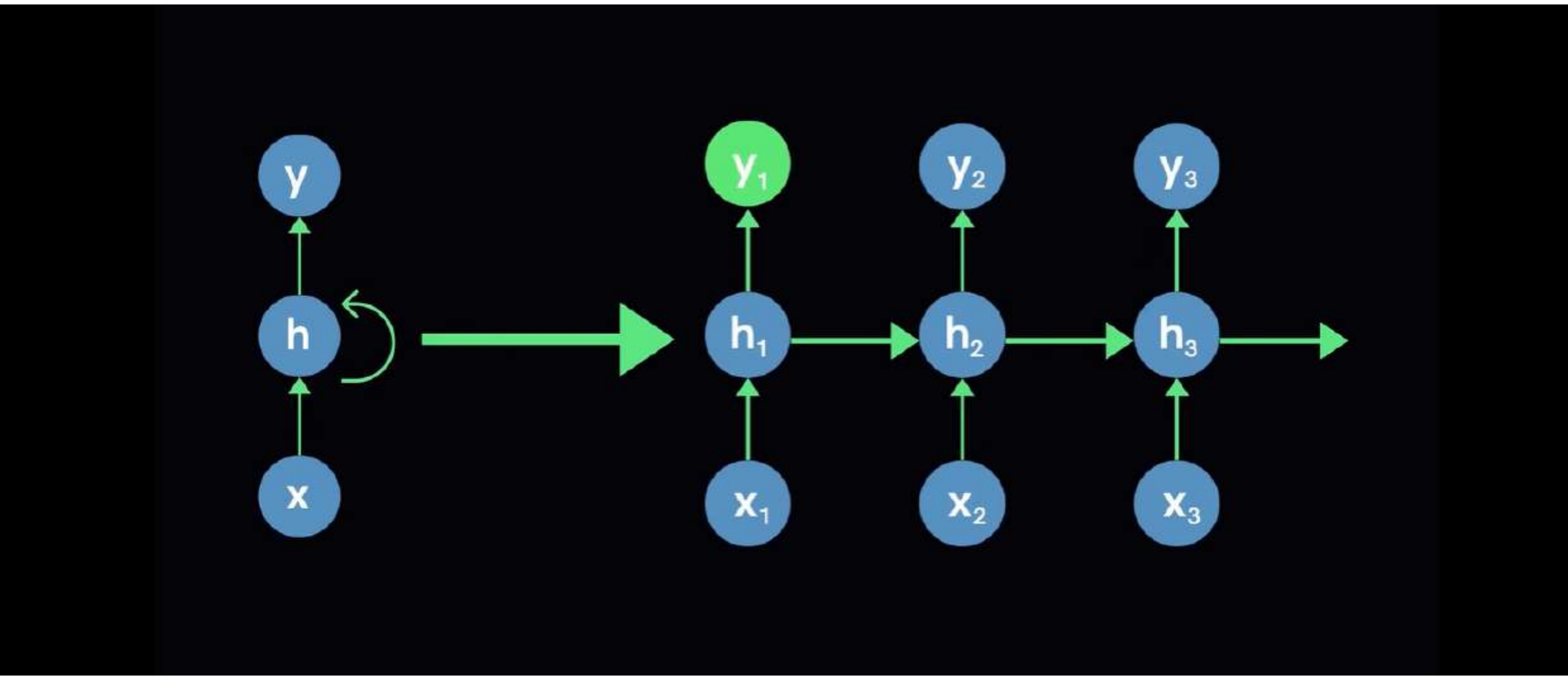
Deal with variable length sequences

Maintain sequence order

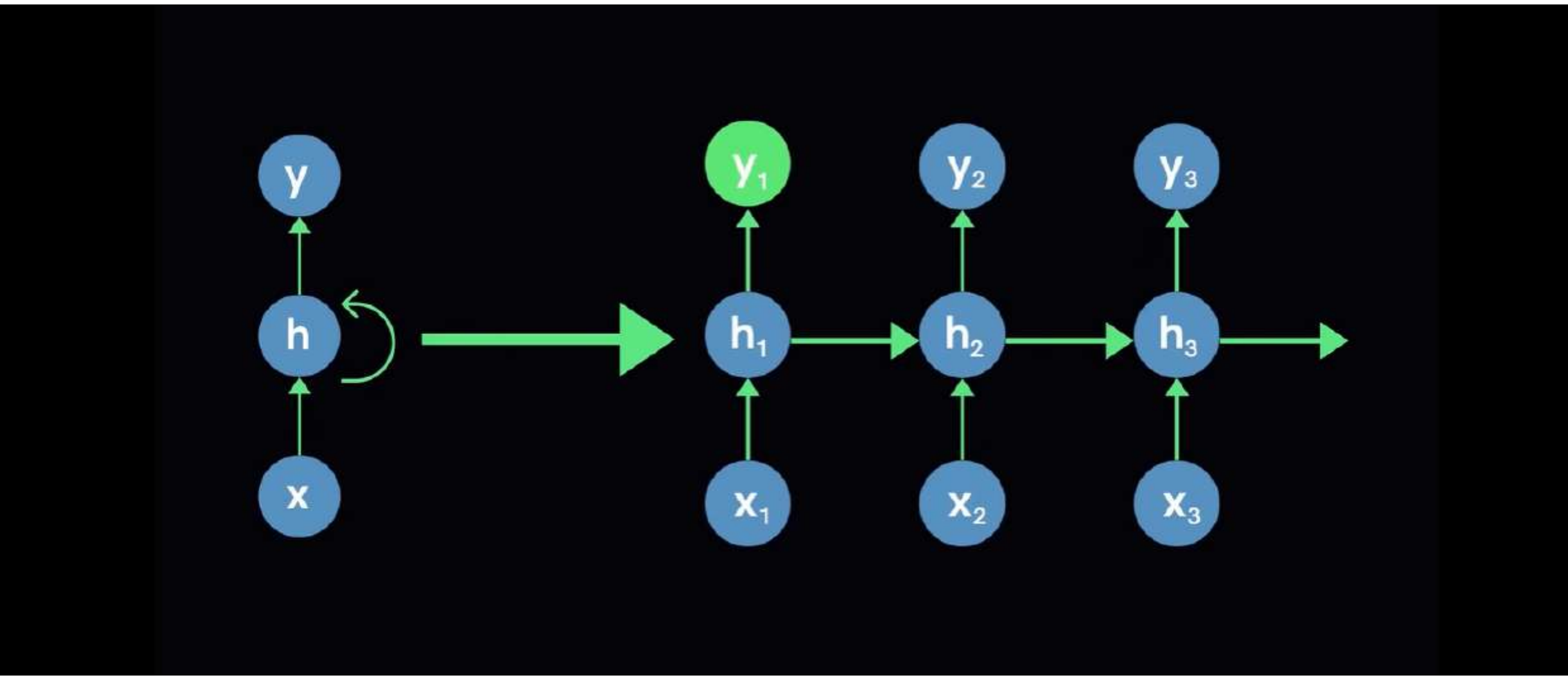
Keep track of long-term dependencies

Share parameters across the sequence


Search
Recurrent Neural Networks
Uses a feedback loop in the hidden layers



y
h
x
y1
y2
y3
h1
h2
h3
x1
x2
x3



y
h
x
y1
y2
y3
h1
h2
h3
x1
x2
x3

Training an RNN

Uses the Backpropagation algorithm

Backprop applied for every sequence data point

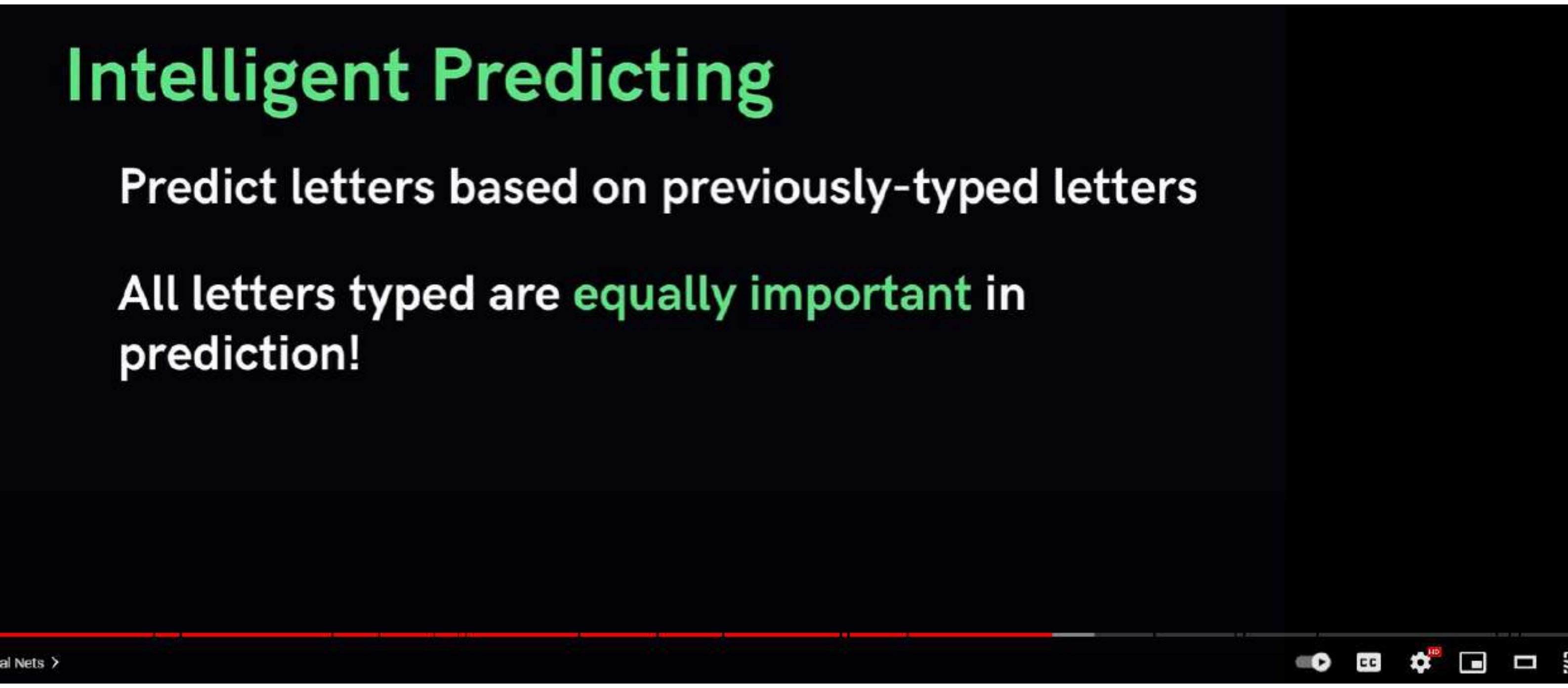
Backpropagation through Time (BTT)

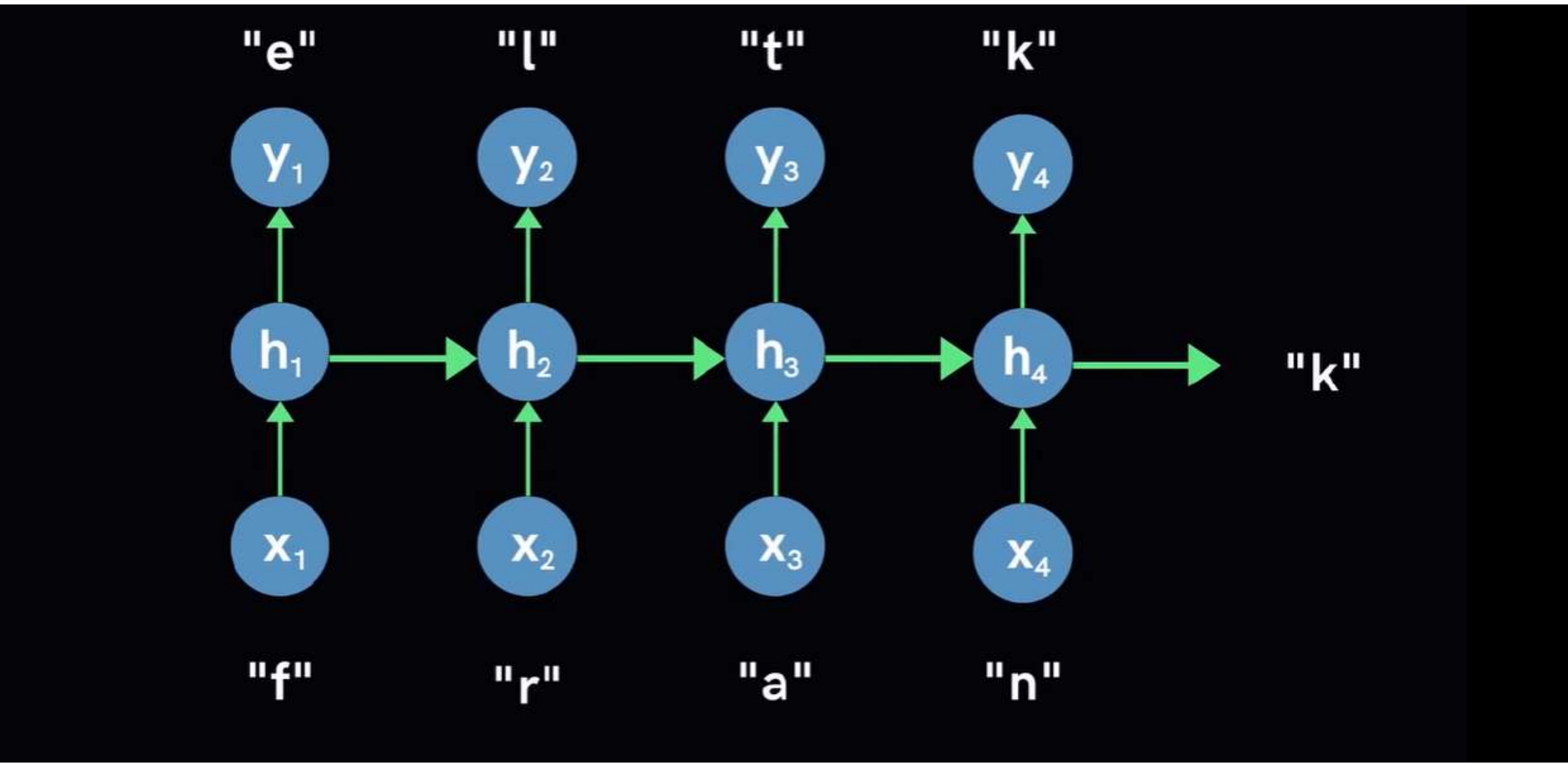
Training an RNN

Uses the Backpropagation algorithm

Backprop applied for every sequence data point

Backpropagation through Time (BTT)


Intelligent Predicting
Predict letters based on previously-typed letters
All letters typed are equally important in prediction!
al Nets


"e"
"l"
"t"
"k"
y1
y2
y3
y4
h1
h2
h3
h4
"k"
x1
x2
x3
x4
"f"
"r"
"a"
"n"

Vanishing Gradient Problem

Short-term memory of an RNN is due to VGP

Due to the nature of Backpropagation

In Backpropagation:

$$\text{Error} = \left(\begin{matrix}\text{Predicted}\\ \text{Output}\end{matrix} - \begin{matrix}\text{Expected}\\ \text{Output}\end{matrix}\right)^2$$

In Backpropagation:

$$\text{Error} = \left(\text{Predicted Output} - \text{Expected Output}\right)^2$$

$$W = w + n\frac{de}{dw}$$

Gradients of a layer are calculated based on the gradients of the previous layer

Gradients of a layer are calculated based on the gradients of the previous layer

If initial gradient is small, adjustments to the subsequent layers will be smaller giving rise to vanishing gradients

Every time step in RNN = A Layer

To train, we use BTT

Gradients used make adjustments to weights and biases

Because of VGP, RNNs are unable to learn long-range dependencies

"It was raining on Tuesday"

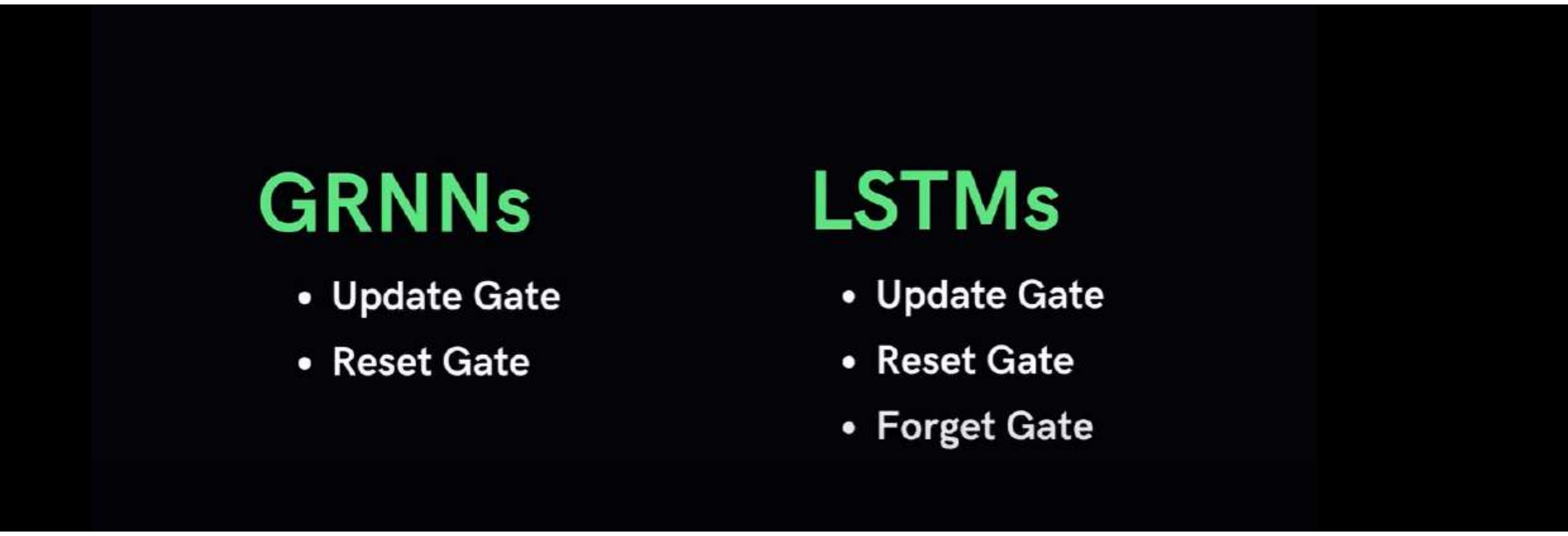
"It" and "was" may not be considered

Model will have to guess based on "on Tuesday"

LSTM - Long Short Term Memory

GRNN - Gated RNN

Capable of learning long-term dependencies using gates


GRNNs
• Update Gate
• Reset Gate
LSTMs
• Update Gate
• Reset Gate
• Forget Gate

Applications of RNNs

1. Natural Language Processing
2. Sentiment Analysis
3. DNA Sequence Classification
4. Speech Recognition
5. Language Translation

Convolutional NNs

- Inspired by the organization of neurons in the visual cortex of the human brain
- Good for processing data like images, audio and video